



# PDF SHARE FORMS

Online, Offline, OnDemand

PDF forms and SharePoint are better together

## PDF Share Forms

## Powershell commands



This guide will describe PDF Share Forms commands available in SharePoint Management Shell.

## Get-PDFShareFormsDocument

### Name:

Get-PDFShareFormsDocument

### Description:

This command allows getting and saving form locally, when form is deployed from PDF template to SharePoint site similarly to getting form from “Get Form” in “PDF Share Forms Enterprise Settings”.

### Syntax:

```
Get-PDFShareFormsDocument [-Web URL] <String> [-List Name] <String> [-Content Type Name] <String> [-Device Type] <String> [-Output Path] <String> [<CommonParameters>]
```

- Web URL <String> – SharePoint Web Site where content type is deployed;
- List Name <String> – SharePoint List where content type is deployed;
- Content Type Name <String> – PDF Share Forms content type which will be used to create document;
- Device Type <String> – Type of device where PDF document will be used, can be Desktop or Mobile;
- Output Path <String> – Location with filename where PDF document will be saved;
- <CommonParameters> – this cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer and OutVariable. More information about these parameters are available in PowerShell help: “get-help about\_commonparameters”.

### Example:

In SharePoint site <http://labs.pdfshareforms.com/sites/labs1> in library called “NewLib” was deployed content type “MyForm”. In following example “Desktop” version of form from this content type will be downloaded to local disk C:\.

Text of PowerShell command:

```
PS C:\Users\Administrator> Get-PDFShareFormsDocument -WebURL  
http://labs.pdfshareforms.com/sites/labs1-ListName “NewLib” -ContentTypeName MyForm -DeviceType  
Desktop -OutputPath C:\MyForm.pdf
```

### Remarks:

This information is also available in SharePoint Management shell:

- Type “get-help Get-PDFShareFormsDocument” to view basic help topic;
- Type “get-help Get-PDFShareFormsDocument -detailed” to view help topic with more details;
- Type “get-help Get-PDFShareFormsDocument -full” to view topic related to technical information about command;
- Type “get-help Get-PDFShareFormsDocument -examples” to view example of using this command.



## Upgrade-PDFShareFormsDocument

### Name:

Upgrade-PDFShareFormsDocument

### Description:

This command allows getting and saving form locally, when form is deployed from PDF template to SharePoint site similarly to getting form from “Get Form” in “PDF Share Forms Enterprise Settings”.

### Syntax:

```
Upgrade-PDFShareFormsDocument [-Identity] <SPListItemPipeBind> [[-TransformDataFunc] <Func<XmlDocument, XmlDocument>>] [<CommonParameters>]
```

- Identity < SPListItemPipeBind > – PDF document item – SPListItem, SPFile or string with document URL;
- TransformDataFunc <Func<XmlDocument, XmlDocument>> – Function (XmlDocument → XmlDocument) that allows to transform old PDF form data before merging with new template version;
- (CommonParameters) – this cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer and OutVariable. More information about these parameters are available in PowerShell help: “get-help about\_commonparameters”.

### Examples:

#### 1. Upgrade PDF Form using SPListItem

At first need to get SPListItem object any way you need (in this example it is done by using list name and ID), then invoke “Upgrade-PDFShareFormsDocument” command on this item.

Text of PowerShell command:

```
C:\PS>$web = Get-SPWeb http://spsiteurl
$list = $web.Lists["ListName"]
$item = $list.GetItemById(130)
$item | Upgrade-PDFShareFormsDocument
```

#### 2. Upgrade PDF Form using document URL

In this example will be used full document URL to upgrade document template.

Text of PowerShell command:

```
C:\PS> Upgrade-PDFShareFormsDocument
"http://yourweburl/LibraryName/PathToYourDocument/documentName.pdf"
```

#### 3. Upgrade Dynamic PDF Form with data transformation

This example shows how to change values inside PDF file using “TransformDataFunc” function, which provides access to [xml] object with document data. Value inside “TextField1” will be changed.



Text of PowerShell command:

```
C:\PS>$item | Upgrade-PDFShareFormsDocument - TransformDataFunc {param($doc)
    $doc.data.Root.TextField1 = "newValue";
    $doc
}
```

#### 4. Upgrade Static PDF Form with data transformation

This example shows how to change values inside static PDF file using "TransformDataFunc" function, which provides access to [xml] object with document data. Value inside "TextField1" will be changed.

Static form data xml looks like

```
<fields>
<field name="Field Name">Field Value</field>
...
<systemFields>
<field name="Field Name">Field Value</field>
...
</systemFields>
</fields>
```

Value of any element can be changed and then updated document can be returned and used instead of initial data.

Text of PowerShell command:

```
C:\PS>$item | Upgrade-PDFShareFormsDocument - TransformDataFunc {param($doc)
    $field = $doc.fields.field | where {$_.name -eq "TextField1"};
    $field.set_InnerXML("newValue");
    $doc
}
```

#### Remarks:

This information is also available in SharePoint Management shell:

- Type "get-help Upgrade-PDFShareFormsDocument" to view basic help topic;
- Type "get-help Upgrade-PDFShareFormsDocument -detailed" to view help topic with more details;
- Type "get-help Upgrade-PDFShareFormsDocument -full" to view topic related to technical information about command;
- Type "get-help Upgrade-PDFShareFormsDocument -examples" to view example of using this command.